Army Research Laboratory

Enhancing S4 with Guidance from the Features of Other Behavior Modeling Systems

By Daniel N. Cassenti and Charneta L. Samms

ARL-TN-0416 **November 2010**

NOTICES

Disclaimers

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5425

ARL-TN-0416 **November 2010**

Enhancing S4 with Guidance from the Features of Other Behavior Modeling Systems

Daniel N. Cassenti and Charneta L. Samms
Human Research and Engineering Directorate, ARL

REPORT DOCUMENTATION PAGE						*Form Approved* *OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**						
1. REPORT DATE *(DD-MM-YYYY)* November 2010		**2. REPORT TYPE** Final				**3. DATES COVERED (From - To)**
4. TITLE AND SUBTITLE Enhancing S4 with Guidance from the Features of Other Behavior Modeling Systems						**5a. CONTRACT NUMBER**
						5b. GRANT NUMBER
						5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Daniel N. Cassenti and Charneta L. Samms						**5d. PROJECT NUMBER**
						5e. TASK NUMBER
						5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: RDRL-HRS-E Aberdeen Proving Ground, MD 21005-5425						**8. PERFORMING ORGANIZATION REPORT NUMBER** ARL-TN-0416
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)						**10. SPONSOR/MONITOR'S ACRONYM(S)** RDRL-HRS-E
						11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.						
13. SUPPLEMENTARY NOTES						
14. ABSTRACT The System-of-Systems Survivability Simulation (S4) is a computer simulation of a military battlefield that was built with the goal of representing realistic Army missions and decision making for the purposes of survivability, lethality, and vulnerability analyses. While S4 is a powerful tool that has the potential to answer important questions about command and control, there are features of S4 that could be enhanced to better represent human behavior. The importance of representing human behavior accurately is evident. S4 tends to idealize human behavior, whereas in the real world cognitive errors occur all the time. In order to be useful to the Army, S4 should represent non-ideal human performance so that it can accurately answer questions about how to improve performance. We make recommendations for updating S4 into a more realistic system for modeling human cognition. We focus on three areas of cognition: perception, memory, and decision making. We derive our recommendations from other modeling tools, including the Adaptive Control of Thought – Rational (ACT-R), the Performance Modulated Functions Server (PMFserv), and the Improved Performance Research Integration Tool (IMPRINT).						
15. SUBJECT TERMS Cognitive Modeling, S4, ACT-4, IMPRINT, PMFserv						
16. SECURITY CLASSIFICATION OF: **Unclassified**			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**		**19a. NAME OF RESPONSIBLE PERSON** Daniel Cassenti
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	UU	36		**19b. TELEPHONE NUMBER** (*Include area code*) (410) 278-5859

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures v

List of Tables v

1. Introduction 1

2. Objective 1

3. Background 2

3.1 Understanding the System-of-Systems Survivability Simulation (S4) 2

3.1.1 Decision Making Agents (DMAs) and Decision Making Processes (DMPs)....2

3.2 Descriptions of Other Behavior Modeling Tools 6

3.2.1 Adaptive Control of Thought – Rational (ACT-R) 6

3.2.2 Performance Modulated Functions Server (PMFserv) 6

3.2.3 Improved Performance Research Integration Tool (IMPRINT) 6

4. Improving the Cognitive Mechanisms of S4 7

4.1 Perception, Memory and Decision Making in S4 7

4.2 Possible Improvements for Perception in S4 8

4.2.1 Object and Self Perception 9

4.2.2 Terrain Perception 10

4.2.3 Summary 11

4.3 Possible Improvements for Memory in S4 12

4.3.1 Division of Memory 12

4.3.2 Forgetting 14

4.3.3 Memory Processes 16

4.3.4 Summary 18

4.4 Possible Improvements for Decision Making in S4 18

4.4.1 Capability Meters 18

4.4.2 Environmental Stressors 19

4.4.3 S4 Productions 20

4.4.4 Utility Values and Utility Learning 20

4.4.5 Summary of Decision-making Recommendations 21

5. Conclusions **22**

6. References **23**

List of Symbols, Abbreviations, and Acronyms **25**

Distribution List **26**

List of Figures

Figure 1. Representation of the platoon DMP. Orange boxes represent the stages, while arrows represent the time course. White boxes represent output of the stages and the orange cylinder represents the storehouse of tasks stored in memory. Adapted from Newton (unpublished). 5

Figure 2. IMPRINT. 7

Figure 3. Example of a visual illusion. 9

List of Tables

Table 1. The nine IMPRINT taxons, their descriptions, and task examples (*19*). 19

INTENTIONALLY LEFT BLANK.

1. Introduction

The System-of-Systems Survivability Simulation (S4) (*1*, *2*) is a computer simulation of a military battlefield that was built with the goal of representing realistic Army missions and decision making for the purposes of survivability, lethality, and vulnerability analyses. While S4 is a powerful tool that has the potential to answer important questions about command and control, there are features of S4 that could be enhanced to better represent human behavior. The importance of representing human behavior accurately cannot be overstated. An inaccurate representation of human behavior tends to idealize human behavior, whereas in the real world cognitive errors occur all the time. In order to be useful to the Army, S4 should represent non-ideal human performance so that it can accurately answer questions about how to improve performance.

The most important goal of any military modeling system is to be as accurate a representation of reality as possible. S4 has sophisticated algorithms and a great deal of promise, but if it does not represent a realistic portrait of the human dimension then it may be interesting, but it will not be useful. That said, S4 is a relatively new system and the programmers must be commended on their system's level of sophistication. There are, however, areas that could use updating. Other modeling tools, such as the Adaptive Control of Thought – Rational (ACT-R) (*3*), the Performance Modulated Functions Server (PMFserv) (*4*, *5*), and the Improved Performance Research Integration Tool (*6*), have sophisticated features that could help to provide more realism of human behavior in S4.

2. Objective

Under the System of Systems Analysis (SoSA) Director's Strategic Initiative (DSI), analysts at the U.S. Army Research Laboratory's (ARL) Human Research and Engineering Directorate were tasked to examine other behavior modeling systems to identify ways S4 may be enhanced to better represent human behavior. Three tools were selected for review. The intention of this report is to document the findings of the tool review and make recommendations on how to enhance the features of S4 using selected features of the three other modeling environments.

3. Background

3.1 Understanding the System-of-Systems Survivability Simulation (S4)

In S4, entities interact in a simulated battlefield. These entities are *platforms*, which represent individual vehicles, Soldiers, and aircraft. Platforms can belong to either *blue force* (typically the focus of the simulation) or *red force* (playing the role of the enemy). Each platform typically includes a crew of personnel and their equipment. Platforms have a location in the environment and a set of capabilities including mobility, communications, sensors, and weapons.

3.1.1 Decision Making Agents (DMAs) and Decision Making Processes (DMPs)

Each combat platform has a Decision Making Agent (DMA) that controls the platform. DMAs have a roster of the initial structure of the force to which it belongs, a list of available capabilities, knowledge of the configuration of its team's networks, a perceptual memory, and a set of Decision Making Processes (DMPs). DMAs range in complexity depending on how many DMPs they have. A single DMP encompasses the set of processes that perform one role, such as communications.

Not all agents in S4 are DMAs. The agents residing in platforms representing civilian crowds or vehicular traffic, for instance, may be much simpler than DMAs. For performance reasons, these simple agents are capable of exhibiting only simple or scripted behavior.

The most common DMA has five DMPs that interact with one another to perceive, process, and execute commands. These five DMPs are the *perception manager*, *report manager*, *communications*, *election*, and *platform* DMPs. The following sections provide a short description of each of these DMPs.

3.1.1.1 Perception Manager DMP

The perception manager is the DMP that receives all perceptions including those from sensors detecting the environment and from communication devices. The perception manager has a *Data Fuser* that combines elements from perceptions into a larger sense of the environment. Its other major function is to produce perceptual memories including changes in the environment and other important perceptions. Perception in the environment is primarily used to detect other platforms, though platforms may also have the ability to sense other entities such as munitions and obscurants. An obscurant may be anything that blocks perception, such as smoke or dust, and indicates the presence or recent presence of a platform. For each potential perception, a sensor must be trained on its place in the environment, and even then there is a probability that governs whether an object is detected.

3.1.1.2 Report Manager DMP

The report manager sends and receives messages regarding situation awareness (SA). The report manager handles three types of reports: *SALocal*, *situation report*, and *spot report*. A SALocal reports the status of the team. A situation report details the status of friendly teams and the spot report details the status of enemy teams. The report manager also monitors memory and sends reports when the memory is important.

3.1.1.3 Communications DMP

The communications DMP receives messages that are sent from the other DMPs in the platform. It also controls the communications devices on the platform, which include voice, data, and hand signal inputs. The communications DMP sends the success or failure of the attempt to transmit a message either into the communications DMP or other DMPs. It also maintains information on the status of any network that may contain useful information for the other DMPs.

3.1.1.4 Election DMP

The election DMP has two functions. It monitors the condition of other platforms in the environment, and it also signals a replacement to fulfill its function should the platform become disabled. The election DMP is meant to ensure continuity of leadership functions, such as command, when damage occurs to the blue forces.

3.1.1.5 Platform DMP

Finally, the platform DMP controls all action completed by the platform. It receives commands transferred by the communications DMP from superiors and must determine how to execute them. The three major types of commands that the platform DMP receives are reconnaissance, maneuvering, and engagement. Reconnaissance and maneuvering include methods of dynamic path finding and formation support. Engagement includes procedures for target selection and weapon selection. In every case, the platform DMP must decide how to manage its sensors, including which sensor or sensors to use. It must also manage its mobility system, weapons, and other devices, not only to fulfill its assigned objectives, but also in the interest of self-preservation.

The most common DMA configuration is a subordinate. Though the five DMPs outlined above are complex in their own right, this type of DMA does not make higher-level decisions; rather, it must receive its commands from a superior platform or the platoon agent.

3.1.1.6 Platoon DMP

The typical agent in S4 has five DMPs, whereas the platoon agent has one more DMP called the *platoon DMP*. The platoon DMP makes plans and decides how to use a network to execute its plans. It is composed of four sets of processes. Each set acts as a stage through which a plan is formed. When the plan reaches the final stage, the plan may then return to the second stage to be reevaluated.

The first stage is the *decision trigger*, which initiates planning. The decision trigger can be a response to a host of triggering events, such as detecting a new enemy platoon. The decision trigger sets into motion *task selection*. Task selection is one cycle through the four sets of processes. The complete set of cycles comprises planning.

The *state builder* is the second stage of the platoon DMP. The purpose of the state builder is to project actions of both blue forces and red forces into the future using possible movement and combat and assumptions about the goals of red forces and its knowledge of blue forces. These possibilities compose a *state tree*. A state tree is composed of nodes and links and represents the passage of time. The first node (i.e., the root) is the current state and subsequent nodes are future states containing state information for relevant blue and red forces (i.e., position, heading, speed, and attrition). A split into two or more branches occurs when the red forces may make more than one choice. The state tree is built until it reaches a *horizon* or a set number of steps when the state builder will no longer build the state tree. One path from the node at the end of the tree (i.e., a leaf) to the root is considered one *state sequence* of the tree.

The third stage is the *mission evaluator*, which evaluates the state tree. First, the mission evaluator analyzes each state sequence and produces two values—the *purpose judgment* (pj), which reflects how well the action state sequence meets the goal and the *unit preservation judgment* (uj), which reflects how much damage the state sequence inflicts on the DMP's platoon. The mission evaluator then uses these values to order the state sequences from best- to worst-case scenario and uses the median of a combination of pj and uj and the worst-case scenario to provide a *state tree evaluation*.

The *task selector* is the last stage and uses a *task template* (i.e., possible tasks) to select tasks that can parameterize new task plans. These parameters include times and routes. These possible plans are then re-processed (i.e., war gamed) through the state builder and mission evaluator to evaluate how well they will work. The task selector selects the most effective task (which could be the current task that has extra positive weight over the other possible tasks to represent inertia). This task proceeds to execution procedures including the path executor, which manages blue force formations and maneuvering, and tasks executors, which assign given tasks to different blue force platforms. Figure 1 represents the platoon DMP.

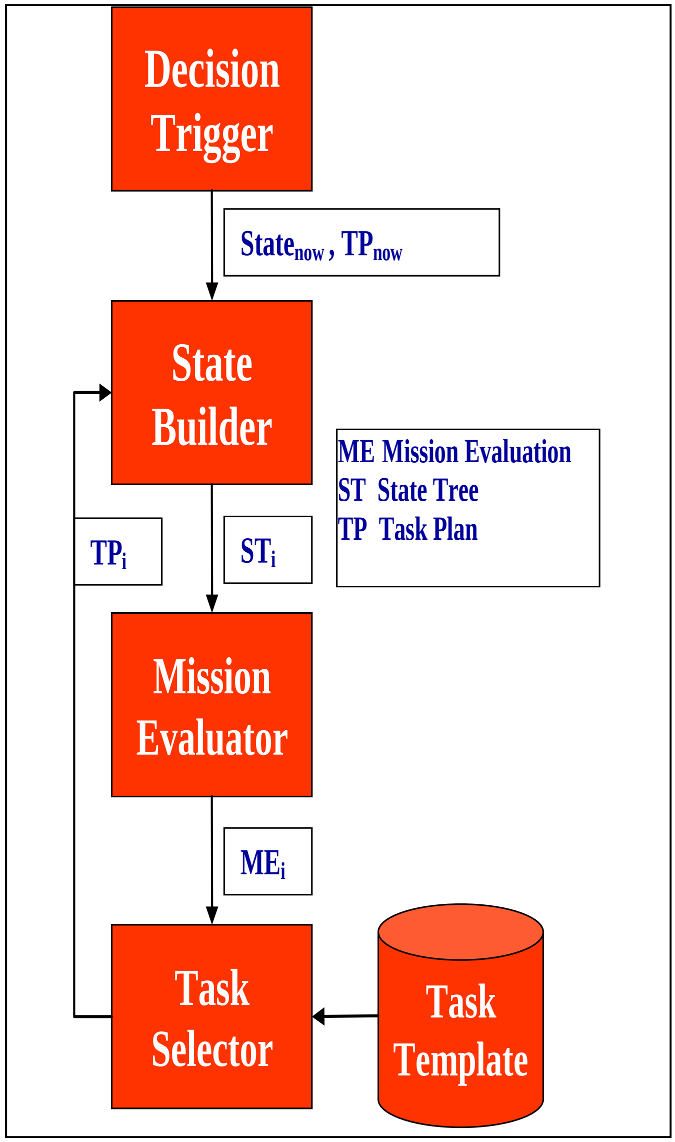

Figure 1. Representation of the platoon DMP. Orange boxes represent the stages, while arrows represent the time course. White boxes represent output of the stages and the orange cylinder represents the storehouse of tasks stored in memory. Adapted from Newton (unpublished).

Outside the scope of this report are the company and battalion agents. These agents outrank the platoon agent and include additional DMPs, but the purpose of this report is to recommend changes to DMPs only up to the platoon agent.

3.2 Descriptions of Other Behavior Modeling Tools

3.2.1 Adaptive Control of Thought – Rational (ACT-R)

The ACT-R cognitive modeling system (*3*) is arguably the most widely used architecture for modeling human cognition. The goal of the system is to be a unified theory of cognition (*7*) or the one system that can model all cognitive processes. Although this goal is lofty and may not be achieved, it provides a basis for always updating and revising ACT-R until all types of mental activity are modeled more or less accurately and precisely in its latest incarnation (*3*, *8*). ACT-R simulates the environment, perceptual processes, cognitive processes, and motor behavior. ACT-R is the primary modeling system for making recommendations to improve S4.

3.2.2 Performance Modulated Functions Server (PMFserv)

The PMFserv (*4*, *5*) is a complex system for modeling that aims to model all aspects of psychology, not just cognition as does ACT-R. In addition to modeling aspects of cognition, PMFserv also models aspects of psychobiology, industrial-organizational psychology, and clinical psychology. PMFserv's modules control different aspects of psychology in order to choose actions for agents that interact with each other in PMFserv's environment. More detail on how PMFserv works is outside the scope of this report. Although PMFserv is an impressive system, much of PMFserv's aspects are not a good fit for S4 due to the fundamental changes in architecture S4 would need to undergo to incorporate these features. However, there is one exception that could be used to augment S4's decision-making processes.

3.2.3 Improved Performance Research Integration Tool (IMPRINT)

The Improved Performance Research Integration Tool (IMPRINT) (*6*) developed by ARL is a dynamic, stochastic, discrete event simulation tool designed to predict the effect of warfighter performance on system performance. Unlike S4, IMPRINT users build unique task network models to represent all of the functions and tasks the human operators must complete for a particular mission to examine issues such as mental overload, mission completion time, and mission success. IMPRINT also provides analysts with the capability to examine the effect of environmental stressors such as heat, cold, noise, and vibration on warfighter and system performance; we recommend that these aspects be used to make changes in S4. Figure 2 depicts a screen capture of IMPRINT.

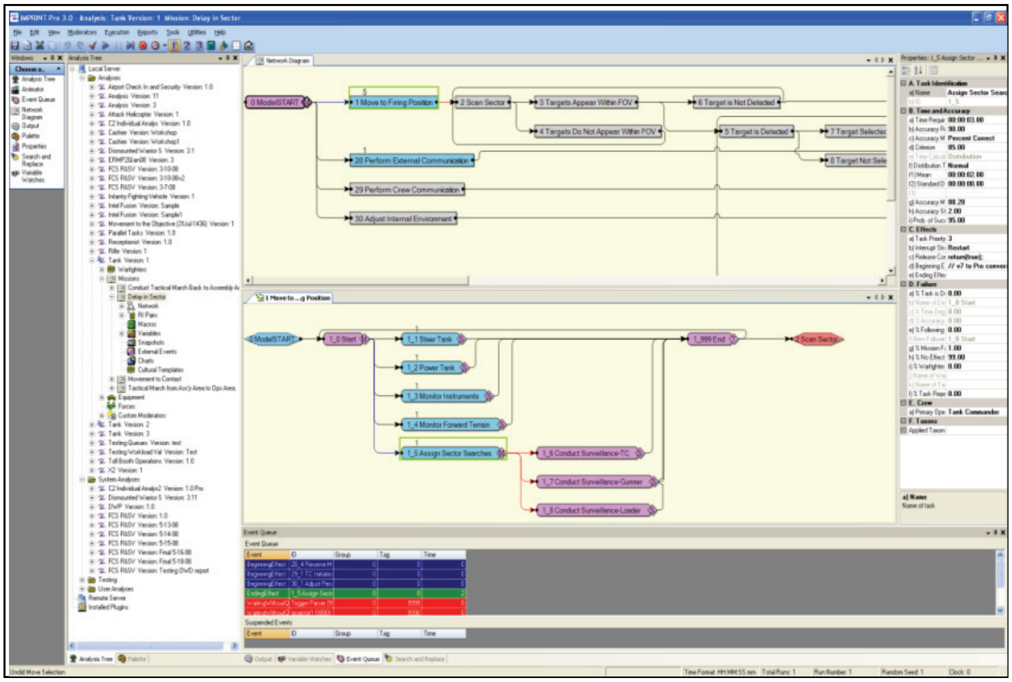

Figure 2. IMPRINT.

4. Improving the Cognitive Mechanisms of S4

Now that we have discussed the structures of S4 agents, we can continue to the goal of this study, which is to improve the cognitive mechanisms of S4. Although the preceding discussion was necessary to lay a foundation for S4, the following discussion focuses on those areas of cognition in which S4 may be improved. These sections discuss in more detail the current cognitive mechanisms behind perception, memory, and decision making.

4.1 Perception, Memory and Decision Making in S4

There are three types of perception in S4: perception of terrain, self, and objects in the environment, including other platforms, communication from other blue force platforms, obscurants, and munitions. The perceptions of terrain and self are terms that must be used loosely, however. Currently, the system's perception of terrain and self is perfect, since it is stored in memory, and thus does not require sensor input. Perception of self, nevertheless, is limited to only certain factors, such as speed and remaining ammunition.

Perception of objects in the environment is guided by an attentional mechanism that is directed by the platform. This type of perception requires visual sensors for visual perception, acoustic sensors for noise detection, and communication sensors for voice communication from other blue force platforms. For visual perception of an object to occur, a sensor must be focused on the field of view where the object is located.

Perception of objects is not perfect like perception of terrain. For visual perception, there is a probability that the visual object will not be detected. The probability is different for each object and changes due to factors such as distance from the observing platform.

Each perception of an object that registers with the platform is stored in memory and can be recalled perfectly (the system does not "forget" information). Although these memories may not be forgotten, they may be updated or deliberately discarded. This aspect allows blue forces to keep a record of the movement of objects and voice communications over time. The platform also has a query mechanism for searching the memory database.

Perceptions are not the only type of memory. Memory also stores a roster of available blue forces and a network that represents how these forces are organized. Plans and goals from the platoon DMP are also stored in memory once they are generated. Memory of the terrain is stored as well, albeit in a different way. The memory of terrain is stored as a collective memory that may be accessed by any platform. The memory of the terrain also represents an interface to the *mobility graph*—a graph that indicates paths for movement for the platforms.

The most notable property of S4's memory is its perfection. Memory is not lost or misrepresented, though the reality of human experience is that memory has mechanisms that make memory imperfect, including decay (i.e., memories weaken over time) and interference (i.e., memories can be confused for one another). This is discussed further in subsequent sections.

The decision-making process was discussed previously in detail in terms of the structures that bring a decision trigger to a final plan. The largest questions remaining regard why the state builder chooses certain red force actions to plan into the future and how the task selector chooses a task (which it then projects into the future by re-routing back to the state builder).

First, the red force actions chosen by state builder are merely drawn from the set of actions that the red forces may do. Although another approach would be to read the possible intention of the red forces and choose possible actions that fit that intention, the state builder instead minimally infers intent on the part of red forces.

Second, the possible tasks that may be chosen by the task selector are drawn from a cache of 6 tactical tasks (though current plans indicate 32 tasks to be programmed) such as block, attack by fire, fix, and others. Some of these tactical tasks are not applicable to a given situation and will not be selected. For example, if the advance of a red force platform triggers planning, attack by fire is a viable possibility, but fix is not, thus limiting the number of tactical tasks that the task selector will consider.

4.2 Possible Improvements for Perception in S4

The perceptual-active cycle (*9*) is a theory of cognition that outlines three stages of cognition that constantly loop as an individual engages with the world. One of these stages is perception, which is the process between sensing and constructing a representation of the world. Perception feeds into cognition where the individual performs mental calculation to both make inferences about the perceptions and decide on a course of action (i.e., decision making). The action is carried out in the third stage and is meant to change an aspect of the environment (e.g., picking up an object). Perception then processes how the environment changed due to the action and a new cycle begins.

Perception is obviously an integral part of human cognition. The question of importance here is how closely S4's perceptual mechanisms match empirical research. At first glance, S4 does incorporate many of the concepts of perception research, including a spot light model of perceptual attention to objects (i.e., attention is directed to objects of interest deliberately; see reference 10). The primary concern, however, is that perception is too "good" in S4.

In humans, perception is rarely perfect, yet in S4, perception is often flawless, including perception of terrain and perception of self. The perception of self does have limitations in what can be sensed, but those categories of perception are automatically perceived and perceived completely accurately.

4.2.1 Object and Self Perception

The perception of objects is not perfect and is limited by whether sensors have the ability to perceive the objects, unlike perception of terrain or the categories of self-perception that may be perceived. If an object is out of a visual sensor's view or an acoustic event is out of an acoustic sensor's range, then the object will not be perceived. In addition, if certain physical properties hinder perception such as distance, there is a probability that the object will not be perceived.

Nevertheless, even in object perception, there is too much perfection. If the object is perceived, it is identified and all the parts of the object that are within view are also perceived accurately. However, human experience indicates that perception is rarely perfect. Illusions are a classic example of mistaken perception, such as in figure 3.

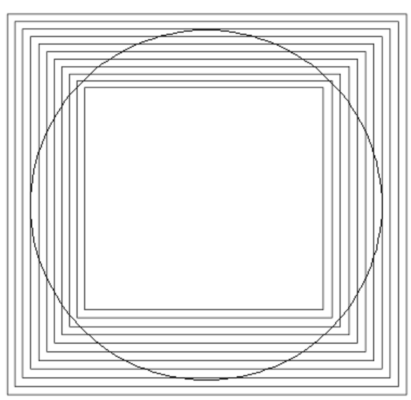

Figure 3. Example of a visual illusion.

The illustration depicted in figure 3 displays a circle superimposed within concentric squares. The circle appears to have flattened sides along its radius though it is perfectly round (*11*). The concentric squares also take on a three-dimensional appearance though they are drawn in two-dimensional space. Both illusions represent erroneous perception illusions represent only one example of erroneous perception. They illustrate a much larger problem with perception—mistaking one object for another. In figure 3, concentric squares may be mistaken for a square-shaped tunnel. A more common example may be mistaking someone in a crowd for a similar-looking acquaintance.

People misperceive objects often, including in the military. In the case of friendly fire, a Soldier mistakes a fellow Soldier or another type of friend for an enemy and fires on that individual. If S4 is to represent reality it must constrain perception by more than just a probability that an object will not be perceived, but also a probability that it will be misperceived.

ACT-R can model misperceptions through similarity values that indicate how much two items are related to one another. S4 developers need to decide how closely related objects may appear (for visual sensors) or sound (for acoustic sensors) like each other. For example, confusing a red force for a blue force may (or may not) be much less likely than confusing specific types of red forces (i.e., a squadron of jeeps versus tanks). In this case, red force and blue force similarity may be set at say 0.1 whereas types of red force may be set 0.8. These similarity values should be included in a table and when an object in the environment is detected that object activates its representation but also adds or subtracts random, normally distributed values to that representation and similar items. The representation with the highest value (called activation in ACT-R) should be the perceived identity of the object. This situation creates infrequently misperceived objects, but does create a mechanism that allows imperfect perception.

Self-perception (albeit to limited information) is also in need of reform. Although many of the measures that are self-perceived may be taken directly from gauges (e.g., speed on a speedometer), there is a chance that these gauges will either malfunction or be misread. A probability should be attached to self-perceptions that these meters may be misread. When a self-perception is needed, the same type of mechanism described above should be used. Since self-perceptions regard quantifiable measures, a table of similarity between numbers should be constructed. Different formats gauges (e.g., meter versus displayed number) may also have different similarity values. For example, a meter gauge may have similar numbers that are close to it on the number line, whereas a number display may have similar numbers that look the same (e.g., 3 and 8).

4.2.2 Terrain Perception

Perception of terrain is also perfect in S4 and this is an even more pressing problem than perfect self-perception. In the cognitive literature, scene perception may be very error prone (e.g., reference 12). In addition, collective knowledge of terrain across all platforms is particularly unrealistic. Even in cases where a map of the terrain exists, all platforms may not have access to the map (and therefore must rely on perception); and even if a map is available, a platform may misinterpret it.

We recommend that S4 developers include provisions for providing or not providing maps to some or all platforms. A probability that the map or perception of terrain is misinterpreted should be included in the set of S4 perceptual mechanisms. If the terrain is misperceived, then damage may occur to the platform when it chooses a bad route or plans may fail such as when the height of a mountain is misperceived and a projectile intended for a red force platform does not clear it.

The idea of a map also brings up the interesting possibility of using external memory (*13, 14*). Like perception, memory is imperfect and therefore people frequently alter the environment (e.g., write a note or put keys by the door) to remind themselves to take a specific action. Environmental changes of this sort are called external memory because instead of relying on fallible internal memory, an individual uses the environment as a type of memory.

Possessing a map is a form of external memory and if implemented in S4 would be a realistic method of implementing similar terrain perception mechanisms that already exist in S4. A consequence of using a map as opposed to direct memory is that platforms must locate themselves on the map as opposed to the first-person perception of terrain, and therefore must interpret the map and translate its features into environmental features.

When a platform locates itself on the map there should be a probability that the self-perception of location is in error (e.g., from global positioning software jamming). If it is, then the platform will locate itself in the wrong portion of the terrain. As the platform moves, perceptual sensors will inevitably notice that expected features of the environment do not match the location on the map. The platform may then attempt to locate itself again on the map. The failure to locate itself on the map, although almost inevitably corrected, is wasted time and if the platoon DMP mistakes the location on the map, then faulty decision making is likely.

The perception of the map must also have a probability that the features of the map could be misinterpreted. The same mechanisms drawn from ACT-R that have guided recommendations on changing S4's perceptual mechanisms in the present section also apply here. There is one exception to this rule, however. Perception of the map should have a higher probability that perception will be mistaken (i.e., greater similarity between perception of objects on a map). We recommend greater similarity values because interpreting a map has two major sources of error—error in perceiving objects and terrain, and errors in interpreting the symbols on the map that represent objects and types of terrain.

4.2.3 Summary

Section 4.2 focused mainly on misperception, albeit implemented through multiple types of perception. Clearly, this change is necessary to make S4 more cognitively plausible. Perceptual change recommendations in S4 face a limit on how much can be drawn from other modeling systems because perception in each of the systems is still in a primitive state. This is not surprising given the extreme complexity of perception, especially visual perception. In the sections on memory and decision-making recommendations, a greater variety of mechanisms to update S4 are recommended.

4.3 Possible Improvements for Memory in S4

S4's memory structure is in need of substantial revisions. As it is implemented, memory is a perfect storehouse of information and all information seems to be stored in the same way. There are three concepts from memory research that would benefit S4. First, there are many different *types* of memories instead of just one. Also, cognitive research indicates that memory is not perfect, indeed *forgetting* occurs all the time. Finally, memory should not be considered a storehouse of information; instead, memory is a *set of mental processes*, in which storage is only one of three major functions. In this section, we review cognitive research in memory and how S4's memory may be improved. These suggestions follow from the ACT-R modeling system and from descriptive models of cognition.

4.3.1 Division of Memory

S4 has memory from different sources including from sensors, self-perception, and the shared terrain map, but it does not seem to have any mechanism to distinguish between different categories of memory. A common distinction between types of memory is between long-term memory (LTM) and short-term memory (STM) or, in some psychologist's perspective, working memory (*15*). LTM is the set of memories that are devoted to long-term storage, whereas STMs are currently activated memories intended for completing a current task that are either retrieved from LTM or recent perceptions.

ACT-R has implemented this distinction in its system. LTMs are stored in two LTM modules (i.e., declarative and procedural memory, which is described below). LTMs may be retrieved and used as needed. When retrieving memories from declarative memory, the memory is copied into a short-term buffer before use. Together, the perceptual and declarative buffers represent ACT-R's STM.

Memory may also be classified as one of three types—procedural, declarative, and goal, as defined by ACT-R. The procedural-declarative memory dichotomy is a classic memory distinction (*16*) and has since adopted by many memory researchers. Procedural memory is memory for how to act including making mental calculations (e.g., adding two numbers) and making movements (e.g., writing the answer to an addition calculation). Declarative memory is memory that may be articulated including sensations, facts, and descriptions. Goal memory is, as the name suggests, memory for goals. Goal memory in ACT-R is treated separately from the other two types of memories, but is itself technically a storehouse of facts. In ACT-R, goal and declarative memory have their own buffers, and procedural and declarative memory have separate long-term stores.

The organization of memory is important to its functionality. By putting all memories into one storehouse, S4 cannot take advantage of the functionality that the organization serves. The LTM-STM distinction allows some memories to be stored more or less permanently in one storage location with infinite capacity (LTM) and memories that are needed at the time to be stored temporarily in another location, where there is limited capacity and therefore fewer items to sort through when the model needs a specific memory.

The distinction between procedural, declarative, and goal memory in ACT-R serves an essential role in the operation of the modeling system. Each ACT-R model begins with the insertion of a goal into the goal buffer. This goal defines what task ACT-R is to accomplish and values of different parameters pertinent to the goal (e.g., how many red forces a platform thinks are in the environment). As the model changes these parameters, they may be updated in the goal.

Procedural memory is the engine that drives ACT-R activity. A single procedural memory is called a production and is an if-then statement. The “if” portion of a production sets out conditions that must be true in order for the production to be used. If those conditions are true, the production becomes a candidate for enacting the actions contained in the “then” portion of the production. An example production may be “if the fire alarm is detected, then exit the building.” A series of productions is meant to (but does not always) move the problem state from an initial state to the satisfaction of a goal.

Declarative memory is the storehouse of facts or, in ACT-R terms, chunks. Each chunk has a name, a category, and any number of properties that describe the name. An example of a chunk might be three, which is a number that is one greater than two and one less than four. In ACT-R, this chunk may be represented as “three ISA number previous two next four.” By representing it this way, ACT-R may use this chunk to count from two to four by working in conjunction with three productions. First, a production would be called that searches for a number fact that has “two” (the production could be made general by drawing this value from a slot in the goal chunk that reflects the current number) in the “previous” slot. Once enacted, the next production would say the name of the number fact that was newly selected and a new production would select the number from the next slot in the current chunk. This example illustrates how productions depend on chunks to move the model through mental steps.

We recommend a similar approach for S4. As of this writing, S4 typically goes through simulation cycles in which one DMP process completes one task in 500 ms. In ACT-R, one cycle is the selection and execution of one production in typically 50 ms (this is the default duration, which the modeler may change). Among other improvements, ACT-R’s approach is an improvement at least in terms of granularity, and therefore, would provide more specificity in response time. If a mechanism to change production times was also included in S4, a new timing tool would be added to help validate models built within S4. One important consideration in recommending changes to S4 is to not recommend changes that would pose too radical a makeover to S4 because this would eliminate or exclude some of the positive aspects of what S4 does. Fully adopting the structure of ACT-R's memory represents this kind of change. Instead, we must make reasonable recommendations to adopt those aspects of ACT-R that may be easily compatible with S4.

The DMP structure in S4 is the defining property of the system. We propose that ACT-R's division of memory into STM and LTM as well as goal, declarative, and procedural should also be established within the DMPs. That is, the functionality of each DMP should be maintained, though reorganized. First, LTM should be a split storehouse of procedural memory and declarative memory with one for each DMP. Instead of a shared memory that makes no distinction between types of memories, S4 should have memory for procedures and memory for facts that do not intermix and are relevant only to the functioning of particular DMPs. The one exception to this rule would be the communications DMP. If information is lacking in a DMP, the communications DMP may convey it. This exception must be in place so that critical information not programmed into a given DMP may be transferred.

The goal and declarative buffers (there is no procedural memory buffer because ACT-R only activates one production at a time) should also be implemented in each of S4's DMPs. This will allow only the goal to be stored in one location while important facts take up another location and are easier to sort through than the declarative LTM. There are also perceptual and motor buffers in ACT-R, but in S4 there are DMPs dedicated to these activities. The perceptual buffers and the motor buffers would exist only in the perceptual and platform DMPs, respectively.

These changes should not be considered a rewriting of the way that S4 currently works. Subordinate DMPs (i.e., those that may receive commands from the platoon DMP) are given instructions to perform (i.e., a goal) and use procedures (i.e., productions) in conjunction with facts (i.e., chunks). As an example, consider that the communications DMP transfers a command from the platoon DMP to the platform DMP instructing the platform DMP to fire on a red force platform. In this example, the platform DMP retains the instruction as a goal to be completed and uses select procedures such as aiming a weapon to enact the goal and sorts through its weapons capabilities to figure out which weapon would best achieve its goal corresponding to search of declarative memory and retaining candidates (i.e., weapons) from a larger group of chunks (e.g., weapons and movement capabilities).

4.3.2 Forgetting

The reorganization of memory in S4 is meant to bring S4 to more traditional models of cognition, so that it may better represent human cognition. In this section, we discuss how incorporating forgetting may achieve the same aim. Though forgetting causes memory to be imperfect, adding forgetting will also make S4 more realistic.

There are two primary ways in which people forget—decay and interference. Decay is the weakening of memory as it is idle in STM (note that decay does not occur in LTM). Interference is confusing memories for one another and the chance that an unintended memory will be used in a task rather than the correct one.

To incorporate both forgetting mechanisms into S4, we recommend that the ACT-R approach to both be adopted. The first step is to co-opt the memory activation approach. ACT-R has an activation value associated with each memory. The activation value defines the strength of a particular memory. Memories in long-term declarative memory and STM buffers all have activation values.

The memories stored in LTM have baseline activation values. The baseline activation is either exclusively determined by the modeler or is set by the user, and then it is strengthened or weakened, respectively, as the memory is or is not retrieved from declarative memory. The latter is caused by an option that may or may not be selected by the modeler.

STMs are either drawn from LTM or perception. When they are drawn from LTM, they are chosen based on their activation value. They gain this activation through the activation of similar memories into a buffer (perceptual or declarative buffers) that pass activation into the memory or when retrieval from declarative memory is requested and elements of that declarative memory are common to the request.

In ACT-R, decay is implemented by a simple formula that deletes activation from memories in the short-term buffers for every production cycle when they are not in use. The rate of decay is set by the modeler and the decay on each step is drawn from a normal distribution. The decay rate removes activation strength from the memory until it goes to the baseline activation level, at which point it is removed from the buffer. Once a memory activation system and the new productions cycle division are in place on S4, implementing this approach to memory decay will be simple.

The ACT-R approach to interference may also be adopted into S4. In section 4.2 on perception changes, we discussed how ACT-R's similarity mechanisms may cause misperception. This same approach can be used to cause interference.

When a request for a chunk is made by a production, ACT-R searches memory for a declarative memory that matches the category of the chunk needed by ACT-R. ACT-R selects one of these chunks by using the activation value. Recall that activation is determined by both its base-rate activation and by whatever activation it receives from similar chunks currently in the declarative, perceptual, and goal buffers. Any number of chunks may pass an activation threshold and of these chunks, one is selected. Interference increases when there are more chunks in declarative memory and especially when there are more chunks that are similar to one another. In this scenario, the wrong yet similar chunk may be selected.

S4 could adopt ACT-R's interference mechanism by using the similarity mechanisms that could also improve S4's perceptual process realism. With this in place and an activation system, the final selection of chunks does not necessarily need to use the threshold part of chunk selection. Although this approach would work, we recommend deviating from ACT-R and selecting the chunk with the highest activation rather than a random one that is above a certain threshold. With the right similarity ratings, interference will occur without the threshold mechanism.

4.3.3 Memory Processes

Memory is more than just a storehouse of information. Instead, memory is a set of mental processes just like any other type of cognition such as perception or problem solving. Cognitive psychologists generally agree that there are three processes of memory: encoding, storage, and retrieval. In this section we recommend how these processes may be incorporated into S4 and how external memory may augment each.

Encoding is the process by which memories are placed into long-term storage. There has been much written about learning, but most of it is outside the scope of this work. If the division of memory into procedural and declarative long-term storage is programmed into S4, then the encoding of procedural and declarative memory must be addressed.

ACT-R encodes new productions through its production compilation mechanism. When ACT-R notices that two or more productions are used repeatedly to accomplish the same goal, it collapses the "if" portion of the first production and the "then" portion of the final production and forms a new production with them (*17*). Production compilation allows a series of productions to take only one step.

Although this is a useful mechanism, it would be difficult for S4 to incorporate the same mechanism in the near future. S4 does not yet have the depth to model all the individual cognitive processes that ACT-R currently does. Instead, S4 handles task modeling at a grosser level. For example, when S4 fires a weapon, it merely sets the coordinates of where the enemy is (provided by the perceptual buffer) and fires. In ACT-R, separate productions would run to focus on the red force, identify it as red force, locate it within the field of view, transfer those coordinates to the goal, aim the weapon to the coordinates in the goal, and fire. Both approaches will fire the weapon, but ACT-R may use production compilation to collapse some of those productions to produce fewer cycles; S4 does not currently recognize all those processes. If and when S4 begins to operate on ACT-R's level of cognitive granularity, then production compilation should be attempted. Until then, S4 should have a prescribed set of productions based on platform capabilities.

The mechanism for chunk encoding in ACT-R is more direct. In ACT-R, there are actions that may be included in productions to clear out the contents of a buffer without using decay. When a buffer is cleared, any declarative memory element including in the goal, declarative, or perceptual buffer that is not already contained in long-term declarative memory may be encoded.

We believe that a similar mechanism in S4 is unnecessary. Alternately, encoding new declarative memory should take one production cycle and any declarative memory may be encoded. This memory should be encoded with a low base rate and may be amended. For example, if a red force platform is detected at one location, it should be encoded; and if it moves, that encoded memory should be updated with the new location. This will retain capabilities that S4 currently has.

S4 is intended to simulate military situations and this implies that it should have all the capabilities that military personnel possess. Although there is no mechanism for it in the modeling systems used here, external memory should be an option for S4 platforms as it is for military personnel, and particularly, the platoon leader. S4 should have the ability to encode memories by writing them down or entering them into a computer. A checkbox option should be added to each platform to indicate whether it has the options. Instead of one short cycle, writing or typing information should take up a longer cycle time than merely remembering.

Storage is the second process of memory. Much of this section has been devoted to the storage of memory, specifically how it should be organized and how STMs decay. One part that was not discussed is external memory storage. External memory has a benefit over internal memory storage in that the memories are preserved intact and cannot be confused or forgotten. However, the drawback of external memory storage is that written or computerized notes can become disorganized. This can be circumvented by having an organizational scheme. For storage of external memory, another option may be added to S4 allowing each platform to have or not have an organizational scheme.

Retrieval is the final memory process to be discussed and its function is to draw a memory from LTM to be used in STM. Retrieval should take one production cycle and is the memory process that is susceptible to interference since retrieval is meant to draw the correct memory, but occasionally draws the wrong one. This is also the process that may be most mitigated by external memory.

Although searching for an external memory may be more difficult, the chance that an individual will draw the wrong memory is lower because the notes are written and therefore represent a permanent recording. If an individual misremembers the grid location of an enemy unit due to interference from similar locations from internal memory, the external memory represents an unaltered recording.

The drawback to retrieving from external memory is that clutter from other notes can make search time slow. This should be mitigated by either using a computer (i.e., a system that usually has some inherent organization) or having an organized scheme for storing notes.

4.3.4 Summary

We recommend that S4 revamp its memory component. Instead of one memory to serve all DMPs (and, in the case of memory for terrain, all platforms), S4 should be divided into STM-LTM components. Within STM, S4 should follow the model example of ACT-R and adopt goal, perceptual, and declarative buffers and within LTM, procedural and declarative memory modules. The division of memory also calls for adoption of a production cycle approach to duration modeling. This will be a starting point for more precisely modeling the cognitive processes that take place in DMPs.

In addition to these changes, S4's memory should become more imperfect by allowing forgetting. Forgetting can take two forms—interference and decay. These may be offset by external memory approaches to memory, though external memory should take more time than internal memory for encoding and retrieval.

4.4 Possible Improvements for Decision Making in S4

Decision making in S4 represents the most sophisticated set of algorithms to be discussed here. Though the level of sophistication is great, there is still room for improvement. For example, when state builder puts together a state tree, it considers actions the red forces could make, rather than those they are likely to make. In the military, however, commanders focus on likely enemy actions and ignore those that could take place, but would not advance the enemy's position. This and other issues with S4 decision making is discussed in this section.

4.4.1 Capability Meters

Although much of the recommendations made so far focus on general psychological research or from ACT-R, the first recommendation for improvement to decision making in S4 comes from a different modeling system—PMFserv (*4*, *5*). PMFserv is an agent-based modeling system, which from a system-level perspective, appears to be more compatible with S4 than ACT-R. However, due to a lack of validation efforts (*18*), relying on PMFserv to improve cognitive realism is risky.

Despite this lack of validation, one of the mechanisms from PMFserv is a good fit for S4. In PMFserv's physiological module, there are meters that track an agent's sleep, energy, and stress levels. As sleep and energy decrease or as stress increases, the meters force the agent into different kinds of behaviors, largely to rectify the situation that is being indicated by the meter. For example, if energy is low, then behaviors that increase energy, such as eating will become more likely than any other action that does not involve increasing energy. Likewise when stress is high, the agent will become more likely to choose behavior that removes the stressor (e.g., fight or flight behaviors).

Adopting the same PMFserv meters into S4 would be inadvisable. PMFserv was designed to generate behavior for agents that represent individuals, whereas S4 was designed to simulate military situations and commander decision making in which the agents are platforms, most of which represent multiple persons. Our recommendation is that S4 develop meters that correspond to groupings of individuals and their associated assets.

Platforms are generally limited by their vehicular capabilities, which include fuel, ammunition, and armor. Instead of PMFserv's sleep, energy, and stress meters, the meters in S4 should be fuel, ammunition, and armor. When the fuel meter is low, the platform should begin choosing behaviors that either lead to refueling the vehicle or do not involve motion. If the ammunition meter is low, then the platform should begin choosing behavior to reload or those that do not involve firing. If the armor meter is low (i.e., there is heavy battle damage), then the platform should favor retreating behaviors.

4.4.2 Environmental Stressors

Another change that could be made to improve decision making in S4 is in the area of environment stressors. IMPRINT uses algorithms to predict the effects of stressors such as heat, cold, noise, and vibration on warfighter performance in a mission environment. This feature provides the capability to add more realism to the mission environment. In order to do this, mission tasks must be broken into taxons. Taxons organize the tasks into skills that are required to accomplish the task. The taxons are weighted to represent how much of the task is focused on a particular skill. IMPRINT has nine taxons, as described in table 1. The taxons are used to determine the weighted impact of each stressor on time and accuracy of the tasks within the model.

Table 1. The nine IMPRINT taxons, their descriptions, and task examples (*19*).

Taxons	**Definitions**	**Examples**
Visual	Requires using the eyes to identify or separate targets or objects	• Seeing something move and then recognizing it as an enemy tank
Numerical	Requires processing arithmetical or mathematical calculations	• Measuring an azimuth on a map with a protractor • Estimating the distance between two points on a map
Cognitive (Problem Solving and Decision	Requires processing information mentally and reaching a conclusion	• Locating a fault in an electrical system after troubleshooting • Selecting the best firing position for a machine gun
Fine Motor Discrete	Requires performing a set of distinct actions in a predetermined sequence mainly involving movement	• Assembly and disassembly of the M-16 rifle • Starting the engine of a truck
Fine Motor Continuous	Requires expending extensive physical effort or exertion to perform an action	• Driving a vehicle • Tracking a moving target
Gross Motor Heavy	Requires expending extensive physical effort or exertion to perform an action	• Lifting an artillery round • Loosening a very tight bolt with a wrench
Gross Motor Light	Requires moving the entire body (i.e., not just the hands) to perform an action without expending	• Getting into a prone firing position • Evacuating a tank
Communication (Read and Write)	Requires either reading text or numbers that are written somewhere or writing text or numbers that can	• Reading a preventive maintenance checklist for a vehicle • Writing a letter home
Communication (Oral)	Requires either talking or listening to another person	• Giving a situation report by radio • Receiving a password from someone while on guard duty

Since IMPRINT environmental stressor algorithms are designed to predict the stressor impact on individual warfighters and not platforms, it is not advisable to adopt these algorithms into S4. However, the concept of modeling the impact of the environment in S4 is ideal. S4 analysts could develop algorithms that affect decisions that are made based on environmental factors. For example, if noise is really high in a particular environment, then communication may be effected and the platforms would opt to use more digital than voice forms of communication or the noise would cause communication messages to be misunderstood or not received.

4.4.3 S4 Productions

In section 4.3, we discussed ACT-R's procedural memory and productions, the elements of procedural memory. We recommended that S4 not completely rearrange itself to accommodate ACT-R's production system. In this section, we discuss how the *if-then structure* of ACT-R's productions should be applied only to the task selector in the platoon DMP, where it would not overhaul S4, but would provide it important utility.

There are currently 6 tasks to choose from in task selector with a potential of 32 tasks that have been conjectured but not implemented. This is similar to typical ACT-R models that have a constrained list of productions from which to select, whereas the number of actions that a platform can take across the other DMPs is extensive. Rewriting all of S4's actions into productions would overhaul S4, but capitalizing on the limited number of tasks in the platoon DMP would add utility.

The utility gained would come in two forms. First, the if-then structure of productions would constrain the tasks that would be considered when the task selector sorts through the tasks, because the conditions in the productions would not always match the environmental conditions. This would reduce interference from non-useful productions and thus reduce the need to loop back from the task selector to the state builder and make decision making quicker and more fluid.

Second, the "then" portion of the production would provide an organized set of instructions to subordinate platforms. Currently, S4 selects a task and sends a more general instruction to the subordinate platforms, such as retreat. Under the production format, the instructions could rely less on allowing the platform to interpret the command and instead send instructions that route the commands to the DMP it means to target with more directed instructions on what to do.

4.4.4 Utility Values and Utility Learning

One of our goals is to avoid recommending changes that will completely revamp S4. The changes should keep the overall structure of S4 largely intact. Currently, S4 uses the stages of decision making in the platoon DMP to make a decision. Our recommendation is to keep each of these steps, but to change how the steps are performed. The description of meters should influence the final stage—the task selector. Only those options that the platform is generally able to perform based on the status of the meters should be considered as possible choices.

In this section, we discuss how ACT-R's utility value system of choosing productions and learning utility values could be used to influence the state builder. In ACT-R, a utility value is a number assigned to each production that reflects the number of successes over the number of attempts at using the production to satisfy goals. As ACT-R uses the productions, an algorithm may update the utility value depending on whether the modeler engages utility learning. The utility value reflects how likely the production will be selected when the conditions of the production match the conditions of the environment, with higher values increasing the likelihood. This system both places a priority system on productions with the same "if" conditions and will improve ACT-R's task performance over runs of the model.

This same type of mechanism may be used for the state builder in S4, but instead of attaching the utility values to productions, they should be attached to actions that red forces may make. Also, instead of updating utility values solely on success, they should reflect the number of times the red force used the action under similar conditions as well as whether the action was successful in achieving a red force goal (e.g., half a point for trying it, another half point for advancing the red force position). The change to reflecting whether the action was tried would reflect red force's past history even more greatly. For example, if action A was tried six times and never succeeds and action B was tried once and never succeeded, the fraction of successes to tries would be equal (i.e., zero), but action A should still be more likely. Adding a kind of utility value to red force actions in this way would increase the platoon DMP's ability to predict behavior and would decrease the number of branches that would need to be constructed if there was also a cap on how high a utility value would have to be before triggering a new branch in the state tree.

One other possibility would be to add utility values to the task selector as well. Though this may provide some utility, the small number of possible action at this point may make utility values irrelevant if they are rewritten to be like productions with "if" conditions. Recall that utility values in ACT-R are only meant to distinguish between the production that have the same "if" conditions. The more limited the pool of productions, the less likely utility values would help.

4.4.5 Summary of Decision-making Recommendations

Although S4 has a sophisticated decision-making process within the platoon DMP, there are areas in which it can be improved. With PMFserv, IMPRINT, and ACT-R we derived four suggestions to improve S4.

First, the meter concept in PMFserv could be applied to the limiting factors that should affect platoon-level decision making, specifically, running out of armor, ammunition, or gasoline. IMPRINT would also be able to help by incorporating its system of stressors into the structure of S4. Third, ACT-R's production structure could improve the functioning of the task selector and make the process of choosing a task simpler. Lastly, we recommend adding values to possible red force actions in the same way that ACT-R uses utility values with its productions. This would help increase S4's ability to anticipate what the red force intends to do. Together, these improvements would help make S4’s decision-making process both more efficient and more realistic.

5. Conclusions

As one of the first examples of a system of systems, S4 has already made great strides to becoming a sophisticated system for Army simulation. As such, it has great potential for expanding the toolset available to military planners to simulate complex battlefields and battlefield decisions that represents a less expensive alternative to the real-world simulation that is used currently.

However, to improve its functionality, we recommend enhancements in the area of human dimension variables to bring more realism to S4.

We focused on three areas of cognition in need of improvement—perception, memory, and decision making. The sources of changes were taken from general psychological research and from three modeling systems—ACT-R, PMFserv, and IMPRINT. These systems represent different and important perspectives on cognitive modeling. Adapting these recommendations into the software of S4 will simultaneously bring S4 into the mainstream discussion of cognitive modeling and simulation as well as bring a great deal of research and development that went into building these modeling system into S4’s modeling system.

6. References

1. Bernstein Jr.; R.; Flores; R.; Starks, M. *Objectives and Capabilities of the System of Systems Survivability Simulation (S4)*; ARL-TN-260; U.S. Army Research Laboratory: Aberdeen Proving Ground, MD, 2006.

2. Starks, M. W.; Flores, R. *New Foundations for Survivability/Lethality/Vulnerability Analysis (SLVA);* ARL-TN-216; U.S. Army Research Laboratory: Aberdeen Proving Ground, MD, 2004.

3. Anderson, J. R.; Lebiere, C. The Atomic Components of Thought, Mahwah, NJ, Erlbaum, 1998.

4. Silverman, B. G. "Toward Realism in Human Performance Simulation." In *The Science and Simulation of Human Performance*, Ness, J. W.; Ritzer, D. R.; Tepe, V., eds., New York: Elsevier, 2004, pp 469-498.

5. *PMFserv*, Version 2.1.0; Edutainiacs LLC: Philadelphia, PA, 2006.

6. Archer, S.; Adkins, R. IMPRINT User's Guide; Prepared for the U.S. Army Research Laboratory: Aberdeen Proving Ground, MD, 1999.

7. Newell, A. Unified Theory of Cognition, Cambridge, MA, Harvard University Press, 1990.

8. Anderson, J. R.; Bothell, D.; Byrne, M. D.; Douglass, S.; Lebiere, C.; Qin, Y. An Integrated Theory of the Mind. *Psychological Review* **2004**, *111*; 1036-1060.

9. Neisser, U. *Cognition and Reality*; W. H. Freeman: San Francisco, CA, 1976.

10. Posner, M. I.; Snyder, C.R.R.; Davidson, B. J. Attention and The Detection of Signals. *Journal of Experimental Psychology: General* **1980**, *109*, 160–174.

11. Hurvich, L. M. *Color Vision;* Sinauer Associates: Sunderland, MA, 1981.

12. Rensink, R. A.; O'Regan, J. K.; Clark, J. J. To See or Not To See: The Need For Attention To Perceive Changes In Scenes. *Psychological Science* **1997**, *8*, 368–373.

13 Intons-Peterson, M. J.; Fournier, J. External and Internal Memory Aids: When and How Often Do We Use Them? *Journal of Experimental Psychology, General* **1986**, *115*, 267-280.

14. Zhang, J. The Nature of External Representations in Problem Solving. *Cognitive Science* **1997**, *21*, 179–217.

15. Baddeley, A. D.; Hitch, G. J. "Working Memory." In *Recent Advances in Learning and Motivation,* G.A. Bower, G. A., ed., **1974**, *8*, 47–89.

16. Anderson, J. R. *The Architecture of Cognition*; Harvard University Press: Cambridge, MA, 1993.

17. Taatgen, N. A.; Lee, F. J. Production Compilation: A Simple Mechanism to Model Complex Skill Acquisition. *Human Factors* **2003**, 45, 61–76.

18. Cassenti, D. N. *Performance Moderated Functions Server's (PMFserv) Military Utility: A Model and Discussion*; ARL-TR-4814; U.S. Army Research Laboratory: Aberdeen Proving Ground, MD, 2009.

19. Allender, L.; Salvi, L.; Promisel, D. Evaluation of Human Performance Under Diverse Conditions Via Modeling Technology. *Proceedings of the Workshop on Emerging Technologies in Human Engineering Testing and Evaluation*, 24-26 June 1997; NATO Technical Proceedings: Brussels, Belgium, 1997.

List of Symbols, Abbreviations, and Acronyms

ACT-R	Adaptive Control of Thought – Rational
ARL	U.S. Army Research Laboratory
DMA	Decision Making Agent
DMPs	Decision Making Processes
DSI	Director's Strategic Initiative
IMPRINT	Improved Performance Research Integration Tool
LTM	long-term memory
PMFserv	Performance Modulated Functions Server
SA	situation awareness
SoSA	System of Systems Analysis
STM	short-term memory
S4	system-of-systems survivability simulation

NO. OF COPIES	ORGANIZATION
1 (PDF only)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA 8725 JOHN J KINGMAN RD STE 0944 FORT BELVOIR VA 22060-6218
1	DIRECTOR US ARMY RESEARCH LAB IMNE ALC HRR 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	DIRECTOR US ARMY RESEARCH LAB RDRL CIM L 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	DIRECTOR US ARMY RESEARCH LAB RDRL CIM P 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	DIRECTOR US ARMY RESEARCH LAB RDRL D 2800 POWDER MILL RD ADELPHI MD 20783-1197

ABERDEEN PROVING GROUND

NO. OF COPIES	ORGANIZATION
1	DIR USARL RDRL CIM G (BLDG 4600)

NO. OF COPIES	ORGANIZATION
1	ARMY RSCH LABORATORY – HRED RDRL HRM A J MARTIN MYER CENTER BLDG 2700 RM 2D311 FORT MONMOUTH NJ 07703-5601
1	ARMY RSCH LABORATORY – HRED RDRL HRM C A DAVISON 320 MANSCEN LOOP STE 115 FORT LEONARD WOOD MO 65473
2	ARMY RSCH LABORATORY – HRED RDRL HRM DI T DAVIS RDRL HRM AJ J HANSBERGER BLDG 5400 RM C242 REDSTONE ARSENAL AL 35898-7290
1	ARMY RSCH LABORATORY – HRED FIELD ELEMENT RDRL HRM DE DR HAWLEY PO BOX 6366 FORT BLISS TX 79906
1	ARMY RSCH LABORATORY – HRED RDRL HRS EA DR V J RICE BLDG 4011 RM 217 1750 GREELEY RD FORT SAM HOUSTON TX 78234-5002
1	ARMY RSCH LABORATORY – HRED RDRL HRM DG K GUNN BLDG 333 PICATINNY ARSENAL NJ 07806-5000

NO. OF COPIES	ORGANIZATION
1	ARMY RSCH LAB–HRED ARMC FIELD ELEMENT RDRL HRM CH C BURNS THIRD AVE BLDG 1467B RM 336 FORT KNOX KY 40121
1	ARMY RSCH LAB–HRED AWC FIELD ELEMENT RDRL HRM DJ D DURBIN BLDG 4506 (DCD) RM 107 FORT RUCKER AL 36362-5000
1	ARMY RSCH LAB–HRED RDRL HRM CK J REINHART 10125 KINGMAN RD FORT BELVOIR VA 22060-5828
1	ARMY RSCH LAB–HRED RDRL HRM AY M BARNES 2520 HEALY AVE STE 1172 BLDG 51005 FORT HUACHUCA AZ 85613-7069
1	ARMY RSCH LAB–HRED RDRL HRM AP D UNGVARSKY POPE HALL BLDG 4709 BCBL 806 HARRISON DR FORT LEAVENWORTH KS 66027-2302
1	ARMY RSCH LAB–HRED RDRL HRM DQ M R FLETCHER NATICK SOLDIER CTR AMSRD NSC WS E BLDG 3 RM 343 NATICK MA 01760-5020
1	ARMY RSCH LAB–HRED RDRL HRM AT J CHEN 12423 RESEARCH PKWY ORLANDO FL 32826
1	ARMY RSCH LAB–HRED RDRL HRM AT C KORTENHAUS 12350 RESEARCH PKWY ORLANDO FL 32826

NO. OF COPIES	ORGANIZATION
1	ARMY RSCH LAB–HRED RDRL HRM AS C MANASCO SIGNAL TOWERS BLDG 29808A RM 303 FORT GORDON GA 30905-5233
1	ARMY RSCH LAB–HRED RDRL HRM CU 6501 E 11 MILE RD MS 284 BLDG 200A 2ND FL RM 2104 WARREN MI 48397-5000
1	ARMY RSCH LAB–HRED FIRES CTR OF EXCELLENCE FIELD ELEMENT RDRL HRM AF C HERNANDEZ 3040 AUSTIN RD RM 221 FORT SILL OK 73503-9043
1	ARMY RSCH LAB– HRED RDRL HRM AV S MIDDLEBROOKS 91012 STATION AVE RM 348 FORT HOOD TX 76544-5073
1	ARMY RSCH LAB–HRED RDRL HRM DW E REDDEN BLDG 4 CL 60 FORT BENNING GA 31905-5400
1	ARMY RSCH LAB–HRED RDRL HRM CN R SPENCER DCSFDI HF HQ USASOC BLDG E2929 FORT BRAGG NC 28310-5000
1 (CD ONLY)	ARMY G1 DAPE MR B KNAPP 300 ARMY PENTAGON RM 2C489 WASHINGTON DC 20310-0300

NO. OF COPIES	ORGANIZATION
	<u>ABERDEEN PROVING GROUND</u>
6	DIR USARL RDRL CIM G S FOPPIANO RDRL HR T LETOWSKI RDRL HRM B J LOCKETT RDRL HRM D D HARRAH RDRL HRS L ALLENDER RDRL HRS D B AMREIN
2	DIR USARL RDRL HRM B CHARNETA SAMMS RDRL HRS E DANIEL CASSENTI BUILDING 459